

Отчет в формате pdf

Times New Roman 14pt, межстрочный интервал 1.0

Отправить отчет на algorithms_itmo@mail.ru

Тема письма: *Имя и фамилия, группа, Задание № X*

Отдельное письмо для каждого задания!

Используйте любой доступный вам язык программирования. Вы можете использовать готовые реализации. Полученные выводы и графический материал должны быть информативными и корректными.

Титульный лист:

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
(УНИВЕРСИТЕТ ИТМО)

Отчет

о выполнении лабораторной работы по заданию № X

“Название лабораторной работы”

по дисциплине «Анализ и разработка алгоритмов»

Работу выполнил:

ст. группы XXXX

Фамилия И.О.

Работу принял:

Чунаев П.В.

Санкт-Петербург

2020

Цель работы

Указывается цель лабораторной работы

Постановка задачи

Формулируются задачи, на решение которых направлена лабораторная работа

Краткая теоретическая часть

Приводятся краткие теоретические сведения, касающиеся содержания лабораторной работы, например, определения, описания алгоритмов, методические подходы к решению поставленных задач (не более 2 стр.).

Результаты

Приводятся результаты решения поставленных задач, в том числе графики и таблицы, а также кратко обсуждаются полученные результаты (не более 4 стр.)

Заключение

Делаются выводы о полученном решении поставленных задач и достижении цели лабораторной работы, дается оценка возможности применения приобретенных навыков и умений на практике

Приложение

Приводятся листинги написанных для выполнения лабораторной работы программ с комментариями

Задание 1. Эмпирический анализ временной сложности алгоритмов

Цель работы

Эмпирический анализ временной сложности алгоритмов

Задачи и методы

Для каждого n от 1 до 2000 произведите по пяти запускам замеры среднего машинного времени исполнения программ, реализующих нижеуказанные алгоритмы и функции. Изобразите на графике полученные данные, отражающие зависимость среднего времени исполнения от n . Проведите теоретический анализ временной сложности рассматриваемых алгоритмов и сравните эмпирическую и теоретическую временные сложности.

I. Сгенерируйте n -мерный случайный вектор $\mathbf{v} = [v_1, v_2, \dots, v_n]$ с неотрицательными элементами. Для \mathbf{v} реализуйте следующие вычисления и алгоритмы:

- 1) $f(\mathbf{v}) = \text{const}$ (постоянная функция);
- 2) $f(\mathbf{v}) = \sum_{k=1}^n v_k$ (сумма элементов);
- 3) $f(\mathbf{v}) = \prod_{k=1}^n v_k$ (произведение);
- 4) полагая, что элементы v – это коэффициенты многочлена P степени $n - 1$, вычислите значение $P(1,5)$ путем прямого (наивного) вычисления $P(x) = \sum_{k=1}^n v_k x^{k-1}$ (т.е. оценивая каждый член по одному) и методом Горнера представления многочлена в виде
$$P(x) = v_1 + x(v_2 + x(v_3 + \dots));$$
- 5) сортировка пузырьком (Bubble Sort) элементов \mathbf{v} ;
- 6) быстрая сортировка (Quick Sort) элементов \mathbf{v} ;
- 7) гибридная сортировка (Timsort) элементов \mathbf{v} .

II. Сгенерируйте случайные матрицы A и B размера $n \times n$ с неотрицательными элементами. Найдите обычное матричное произведение для A и B .

III. Опишите структуры данных и методы проектирования, которые использованы в рассматриваемых алгоритмах.

Задание 2. Алгоритмы безусловной нелинейной оптимизации. Прямые методы

Цель работы

Применение прямых методов (одномерные методы перебора, дихотомии, золотого сечения; многомерные методы перебора, Гаусса, Нелдера-Мида) в задачах безусловной нелинейной оптимизации

Задачи и методы

I. Используйте одномерные методы перебора, дихотомии и золотого сечения для приближенного (с точностью $\varepsilon = 0,001$) поиска $x: f(x) \rightarrow \min$ для следующих функций и областей допустимых значений:

- $f(x) = x^3, x \in [0, 1];$
- $f(x) = |x - 0,2|, x \in [0, 1];$
- $f(x) = x \sin \frac{1}{x}, x \in [0,01, 1].$

Подсчитайте количество вычислений функции f и количество произведенных итераций для каждого метода и проведите анализ полученных результатов. Объясните различия в полученных результатах, если таковые имеются.

II. Сгенерируйте случайные значения $\alpha \in (0, 1)$ и $\beta \in (0, 1)$. С использованием этих значений сгенерируйте массив зашумленных данных (x_k, y_k) для $k = 0, \dots, 100$ по следующему правилу:

$$y_k = \alpha x_k + \beta + \delta_k, \quad x_k = \frac{k}{100},$$

где $\delta_k \sim N(0, 1)$ – значения случайной величины со стандартным нормальным распределением. Аппроксимируйте полученные данные линейной и рациональной функциями:

- $F(x, a, b) = ax + b$ (линейная аппроксимирующая функция);
- $F(x, a, b) = \frac{a}{1+bx}$ (рациональная аппроксимирующая функция),

с помощью метода наименьших квадратов путем численной (с точностью $\varepsilon = 0,001$) минимизации функции

$$D(a, b) = \sum_{k=0}^{100} (F(x_k, a, b) - y_k)^2.$$

Для решения задачи минимизации используйте методы перебора, Гаусса и Нелдера-Мида. При необходимости самостоятельно задайте начальные приближения и прочие параметры методов. На графиках (**отдельно для каждой аппроксимирующей функции**) изобразите массив данных и графики аппроксимирующих функций, полученных с помощью указанных методов численной оптимизации. Проведите анализ полученных результатов (в терминах количества итераций, точности, числа вычислений функции и пр.).

Задание 3. Алгоритмы безусловной нелинейной оптимизации. Методы первого и второго порядка

Цель работы

Применение методов первого и второго порядка (градиентный спуск, метод сопряженных градиентов, метод Ньютона и алгоритм Левенберга-Марквардта) в задачах безусловной нелинейной оптимизации

Задачи и методы

Сгенерируйте случайные значения $\alpha \in (0, 1)$ и $\beta \in (0, 1)$. С использованием этих значений сгенерируйте массив зашумленных данных (x_k, y_k) для $k = 0, \dots, 100$ по следующему правилу:

$$y_k = \alpha x_k + \beta + \delta_k, \quad x_k = \frac{k}{100},$$

где $\delta_k \sim N(0, 1)$ – значения случайной величины со стандартным нормальным распределением. Аппроксимируйте полученные данные линейной и рациональной функциями:

- $F(x, a, b) = ax + b$ (линейная аппроксимирующая функция);
- $F(x, a, b) = \frac{a}{1+bx}$ (рациональная аппроксимирующая функция),

с помощью метода наименьших квадратов путем численной (с точностью $\varepsilon = 0,001$) минимизации функции

$$D(a, b) = \sum_{k=0}^{100} (F(x_k, a, b) - y_k)^2.$$

Для решения задачи минимизации используйте градиентный спуск, метод сопряженных градиентов, метод Ньютона и алгоритм Левенберга-Марквардта. При необходимости самостоятельно задайте начальные приближения и прочие параметры методов. На графиках (**отдельно для каждой аппроксимирующей функции**) изобразите массив данных и графики аппроксимирующих функций, полученных с помощью указанных методов численной оптимизации. Проведите анализ полученных результатов (в терминах количества итераций, точности, числа вычислений функции и пр.) и сравните то, что получилось, с результатами работы алгоритмов из части II Задания 2 для того же набора данных.

Задание 4. Алгоритмы безусловной нелинейной оптимизации. Стохастические и метаэвристические алгоритмы

Цель работы

Использование стохастических и метаэвристических алгоритмов (имитация отжига, дифференциальная эволюция, метод роя частиц) в задачах безусловной нелинейной оптимизации и их экспериментальное сравнение с алгоритмами Нелдера-Мида и Левенберга-Марквардта

Задачи и методы

Сгенерируйте зашумленные данные (x_k, y_k) , где $k = 0, \dots, 1000$, следующим образом:

$$y_k = \begin{cases} -100 + \delta_k, & f(x_k) < -100, \\ f(x_k) + \delta_k, & -100 \leq f(x_k) \leq 100, \\ 100 + \delta_k, & f(x_k) > 100, \end{cases} \quad x_k = \frac{3k}{1000},$$
$$f(x) = \frac{1}{x^2 - 3x + 2},$$

где $\delta_k \sim N(0, 1)$ – значения случайной величины со стандартным нормальным распределением. Аппроксимируйте полученные данные рациональной функцией

$$F(x, a, b, c, d) = \frac{ax + b}{x^2 + cx + d}$$

с помощью метода наименьших квадратов путем численной минимизации функции

$$D(a, b, c, d) = \sum_{k=0}^{1000} (F(x_k, a, b, c, d) - y_k)^2.$$

Для решения задачи минимизации используйте алгоритм Нелдера-Мида, алгоритм Левенберга-Марквардта и **хотя бы два** из методов: имитация отжига, дифференциальная эволюция или метод роя частиц. При необходимости задайте начальные приближения и другие параметры методов. Используйте $\varepsilon = 0,001$ в качестве разрешенной ошибки; допускается не более 1000 итераций. Визуализируйте данные и **все** аппроксимирующие кривые, полученные указанными методами численной оптимизации, на **одном** графике. Проведите анализ полученных результатов (в терминах количества итераций, точности, числа вычислений функции и пр.).

Задание 5. Алгоритмы на графах. Введение в графы и основные алгоритмы на графах

Цель работы

Использование различных представлений графов и основных алгоритмов на графах (поиск в глубину и поиск в ширину)

Задачи и методы

I. Сгенерируйте случайную матрицу смежности для простого неориентированного невзвешенного графа со 100 вершинами и 200 ребрами (обратите внимание, что матрица должна быть симметричной и содержать только 0 и 1 в качестве элементов). Преобразуйте матрицу смежности в список смежности. Визуализируйте граф и выведите несколько строк матрицы смежности и списка смежности. Для каких целей удобнее использовать каждое из представлений?

II. Используйте поиск в глубину, чтобы найти связанные компоненты графа, и поиск в ширину, чтобы найти кратчайший путь между двумя случайными вершинами. Проанализируйте результаты.

III. Опишите структуры данных и методы проектирования, которые использованы в рассматриваемых алгоритмах.

Задание 6. Алгоритмы на графах. Алгоритмы поиска пути на взвешенных графах

Цель работы

Использование алгоритмов поиска пути на взвешенных графах (алгоритмы Дейкстры, A и Беллмана-Форда)*

Задачи и методы

I. Сгенерируйте случайную матрицу смежности для простого неориентированного взвешенного графа из 100 вершин и 500 ребер с назначенными случайными положительными целыми весами (обратите внимание, что матрица должна быть симметричной и содержать только 0 и веса в качестве элементов). Используйте алгоритмы Дейкстры и Беллмана-Форда, чтобы найти кратчайшие пути между случайной начальной вершиной и другими вершинами. Измерьте время, необходимое для поиска путей каждому алгоритму. Повторите эксперимент 10 раз для одной и той же начальной вершины и рассчитайте среднее время, необходимое для поиска путей каждому алгоритму. Проанализируйте результаты.

II. Создайте сетку 10x10 с 30 ячейками-препятствиями. Выберите две случайные разрешенные ячейки и найдите кратчайший путь между ними, используя алгоритм A*. Повторите эксперимент 5 раз с другой случайной парой ячеек. Проанализируйте результаты.

III. Опишите структуры данных и методы проектирования, которые использованы в рассматриваемых алгоритмах.

Задание 7. Алгоритмы на графах. Инструменты для анализа сетей

Цель

Использование ПО Gephi для анализа сетей

Задачи и методы

1. Загрузите и установите Gephi с <https://gephi.org/>.
2. Выберите сеть в базе данных <https://snap.stanford.edu/data/> с числом узлов не более 10000. Вы можете выбрать тип и тематику сети по собственному усмотрению (не/взвешенная, не/ориентированная).
3. При необходимости измените формат данных на тот, с которым работает Gephi (.csv, .xls, .edges и т.д.).
4. Импортируйте и обработайте данные сети в Gephi. Проверьте корректность импорта данных.
5. Получите раскладку двух разных типов для графа сети.
6. Рассчитайте доступные показатели сети в разделе Статистика Gephi.
7. Проанализируйте результаты для выбранной сети.

Выполняя работу, делайте скриншоты основных шагов и вставляйте в отчет с комментариями.

Задание 8 (расширенное). Практический анализ современных алгоритмов

Цель работы

Практический анализ современных алгоритмов

Используйте следующую книгу (или ее русскоязычную версию):

Thomas H. Cormen Charles E. Leiserson Ronald L. Rivest Clifford Stein Introduction to Algorithms Third Edition, 2009 (or other editions).

Разделы:

I Foundations

4 Divide-and-Conquer

5 Probabilistic Analysis and Randomized Algorithms

VI Graph Algorithms

23 Minimum Spanning Trees

25 All-Pairs Shortest Paths

26 Maximum Flow

IV Advanced Design and Analysis Techniques

15 Dynamic Programming

16 Greedy Algorithms

VII Selected Topics

Задание для студентов, не имеющих опыта анализа алгоритмов:

I. Выберите **два** алгоритма (интересных вам и **не рассмотренных** в курсе) из вышеперечисленных разделов книги.

II. Проанализируйте выбранные алгоритмы с точки зрения временной и пространственной сложности, используемых методов проектирования и т.д. Реализуйте алгоритмы и проведите несколько экспериментов. Проанализируйте результаты.

Задание для студентов, имеющих опыт анализа алгоритмов:

I. Выберите **один** алгоритм (интересный вам и не рассмотренный в курсе) из вышеперечисленных разделов книги.

II. Выберите **еще один** интересный вам алгоритм, предложенный не более 10 лет назад в исследовательской работе для решения определенной практической задачи (включая алгоритмы оптимизации, алгоритмы на графах и т.д.).

III. Проанализируйте выбранные алгоритмы с точки зрения временной и пространственной сложности, используемых методов проектирования и т.д. Реализуйте алгоритмы (или используйте существующие реализации из исследовательской статьи) и проведите несколько экспериментов. Ваши

эксперименты должны отличаться от экспериментов в исследовательской статье. Проанализируйте результаты.

Дополнительные темы (по указанной книге или ее русскоязычной версии):

II Sorting and Order Statistics

III Data Structures

V Advanced Data Structures

VII Selected Topics

34 NP-Completeness